

(12) UK Patent Application (19) GB (11) 2 310 101 (13) A

(43) Date of A Publication 13.08.1997

(21) Application No 9701471.6

(22) Date of Filing 24.01.1997

(30) Priority Data

(31) 08599695 (32) 09.02.1996 (33) US

(71) Applicant(s)

International Business Machines Corporation

(Incorporated in USA - New York)

Armonk, New York 10504, United States of America

(72) Inventor(s)

Elliot Neil Linzer

Roderick Michael Peters West

Peter Hans Westerink

(74) Agent and/or Address for Service

S R Davies

IBM UK Ltd, Intellectual Property Dept, Hursley Park,
Winchester, Hampshire, SO21 2JN, United Kingdom

(51) INT CL⁶

H04N 7/26

(52) UK CL (Edition O)

H4F FD1B9 FD12X FD3C FD3D FD3P FD3T FD30K

FD30R FD32 FRW

U1S S2213

(56) Documents Cited

GB 2295255 A

(58) Field of Search

UK CL (Edition O) H4F FRC FRD FRG FRM FRP FRR

FRS FRT FRW FRX

INT CL⁶ H04N 7/00 7/24 7/26 7/30 7/32 7/34 7/36 7/48

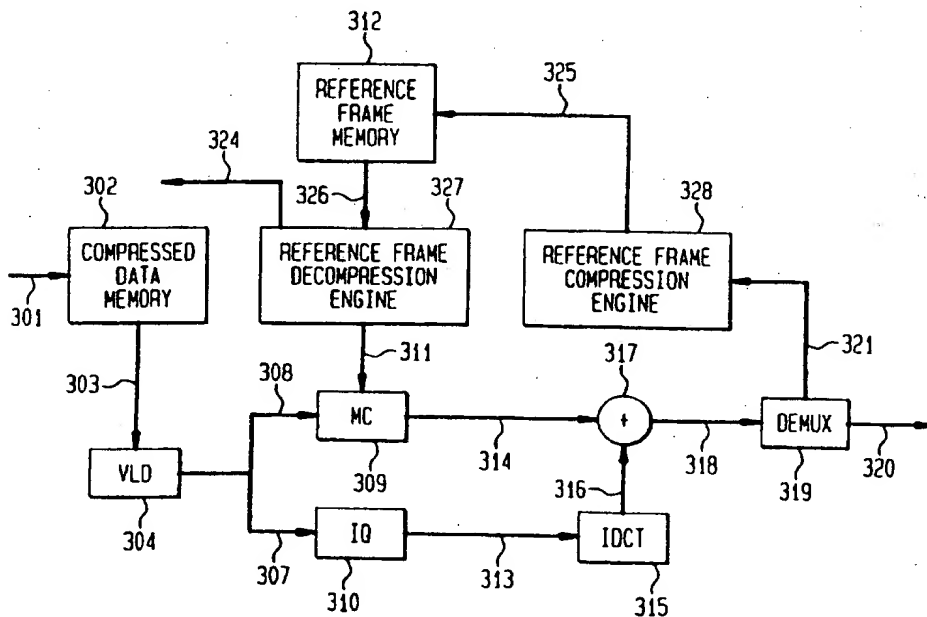
7/50 11/00 11/02 11/04

Online:WPI

(54) Decoding a digital video signal

(57) The memory requirements of a video decoding subsystem are reduced by storing reference frames in compressed form. When the reference frame is needed for motion compensation for a subsequent frame, it is made available by decompressing the stored compressed frame.

FIG. 3



BEST AVAILABLE COPY

GB 2 310 101 A

FIG. 1
(PRIOR ART)

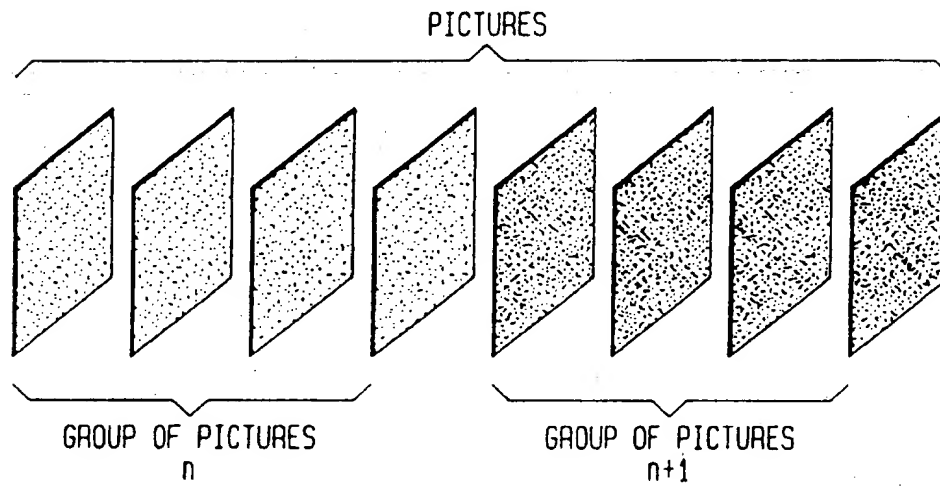


FIG. 2
(PRIOR ART)

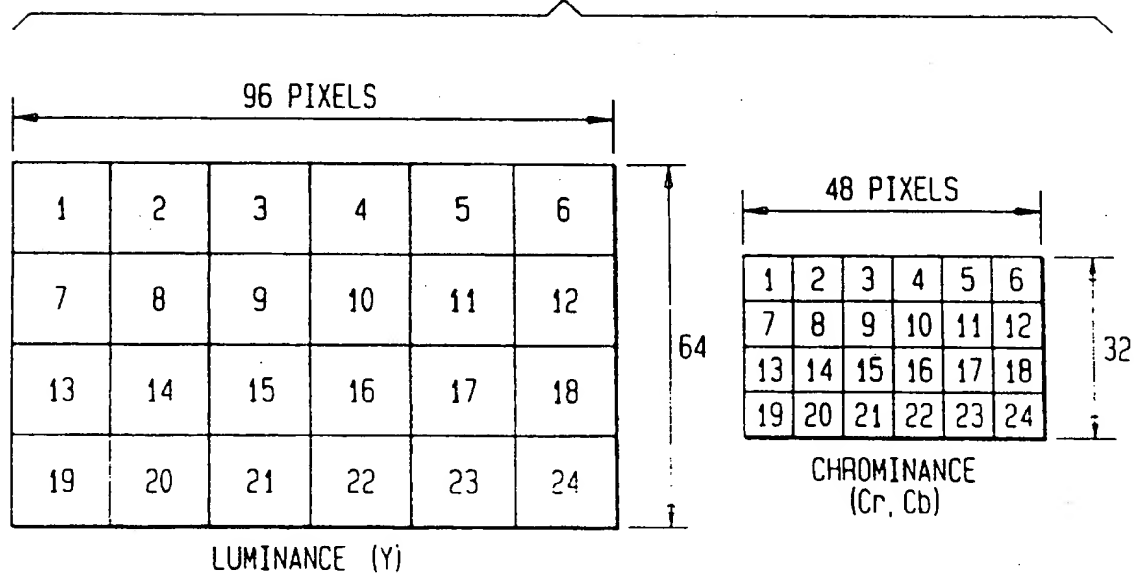


FIG. 3

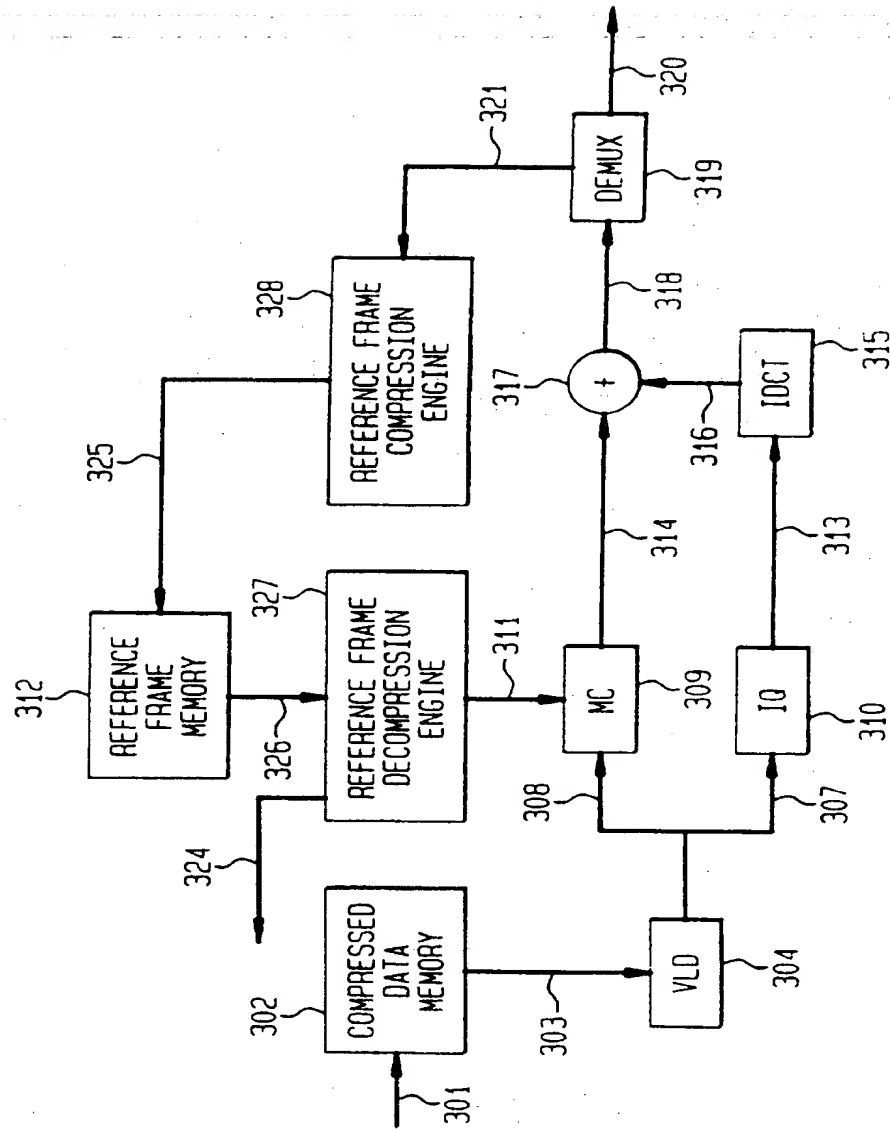


FIG. 4
(PRIOR ART)

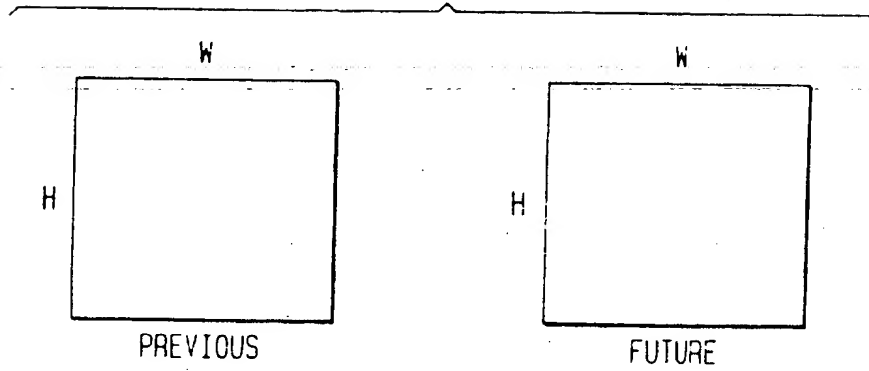


FIG. 5

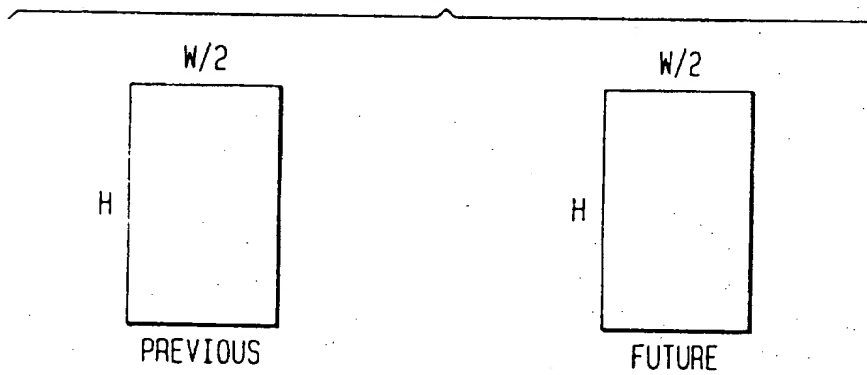


FIG. 6

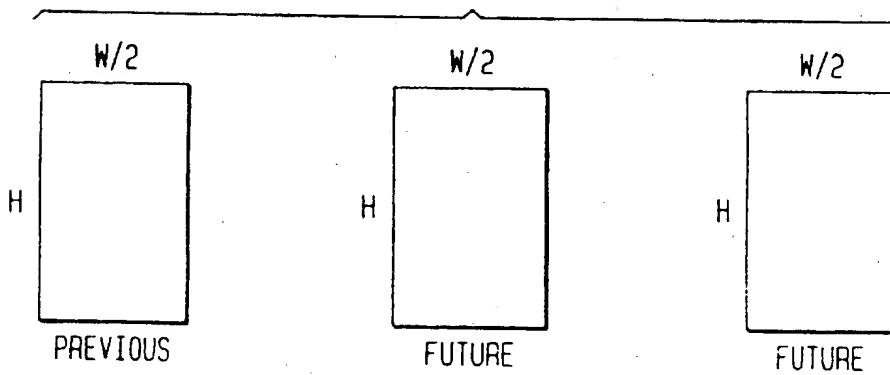


FIG. 7

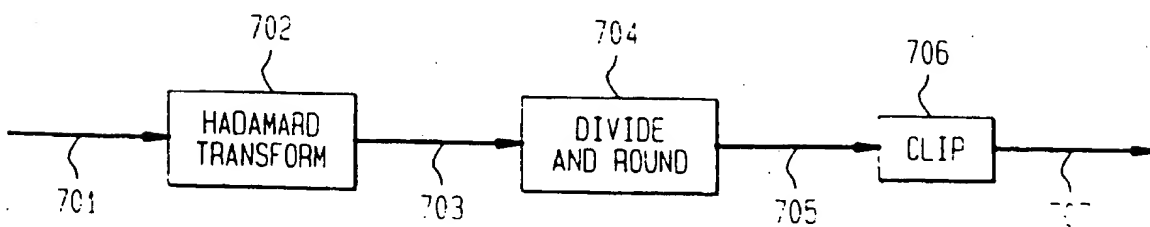


FIG. 8

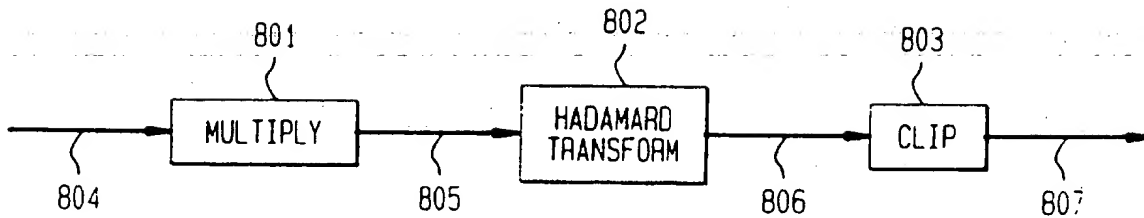


FIG. 9

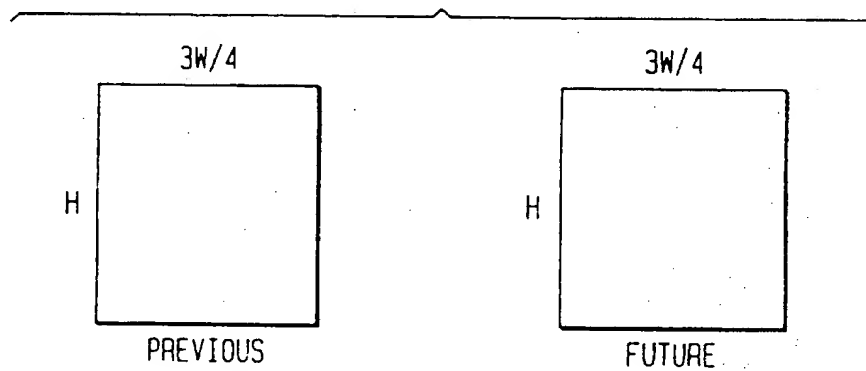


FIG. 10

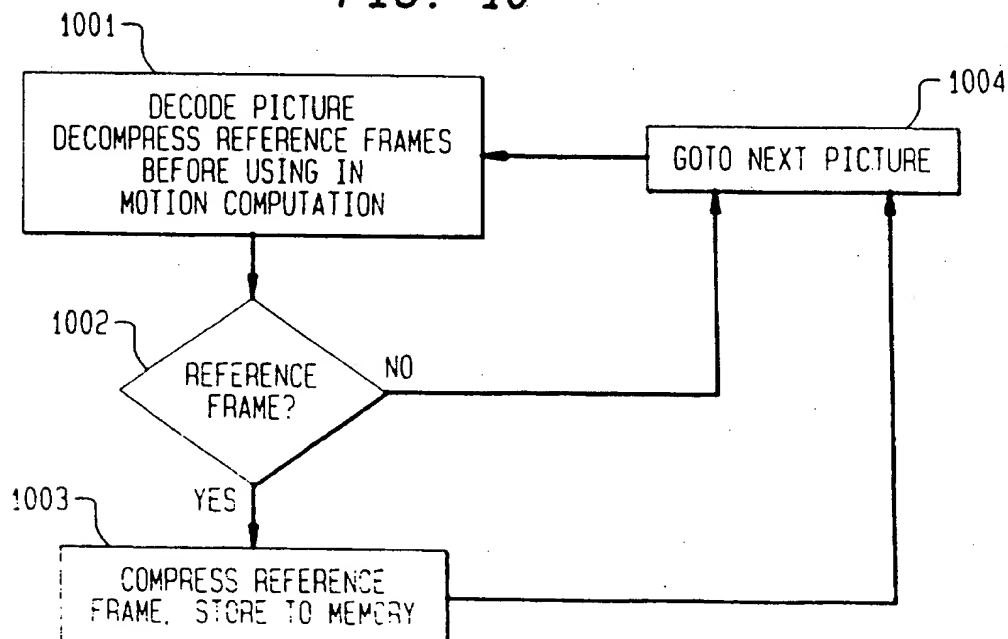


FIG. 11

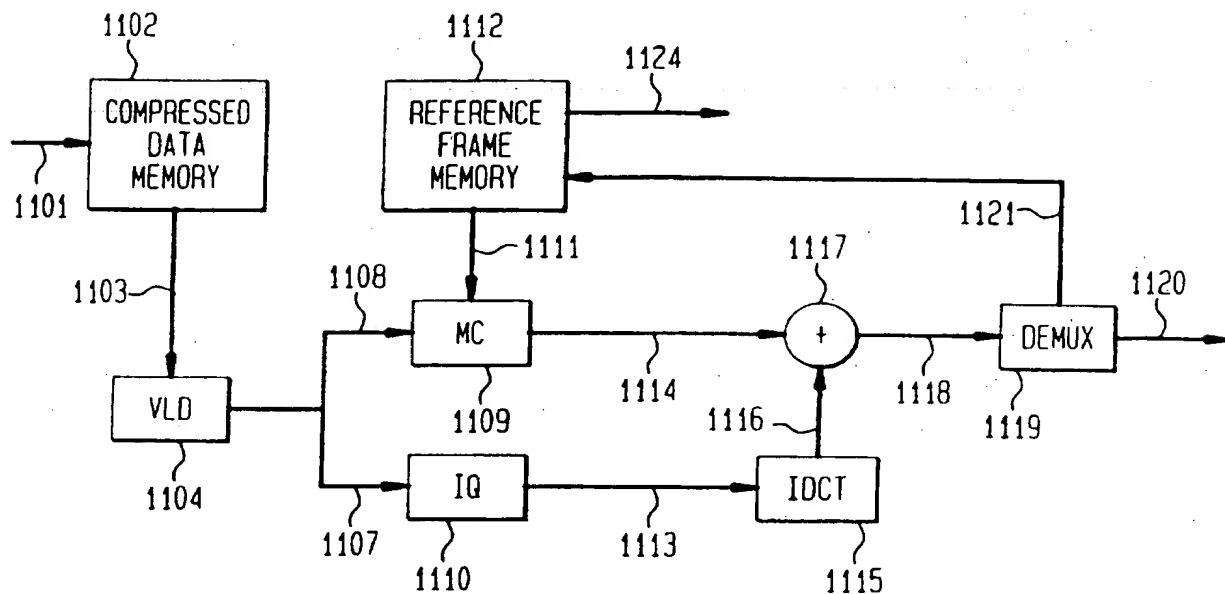
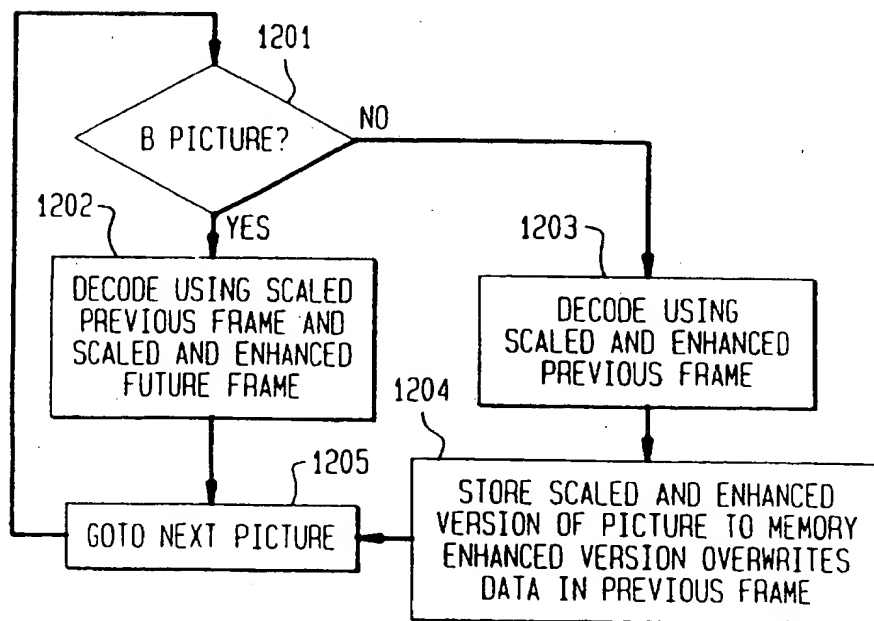


FIG. 12



METHOD AND APPARATUS FOR DECODING A DIGITAL VIDEO SIGNAL

5 The present invention relates to the field of data compression and, more particularly, to a system and techniques for decoding and decompressing digital motion video signals.

10 Technological advances in digital transmission networks, digital storage media, Very Large Scale Integration devices, and digital processing of video and audio signals are converging to make the transmission and storage of digital video economical in a wide variety of applications. Because the storage and transmission of digital video signals is central to many applications, and because an uncompressed representation of a video signal requires a large amount of storage, the use of digital video compression techniques is vital to this advancing art. In this regard, several international standards for the compression of digital video signals have emerged over the past decade, with more currently under development. These standards apply to algorithms for the transmission and storage of compressed digital video in a variety of applications, including: video-telephony and teleconferencing; high quality digital television transmission on coaxial and fiber-optic networks as well as broadcast terrestrially and over direct broadcast satellites; and in interactive multimedia products on CD-ROM, Digital Audio Tape, and Winchester disk drives.

25 Several of these standards involve algorithms based on a common core of compression techniques, e.g., the CCITT (Consultative Committee on International Telegraphy and Telephony) Recommendation H.120, the CCITT Recommendation H.261, and the ISO/IEC MPEG-1 and MPEG-2 standards. The MPEG algorithms have been developed by the Moving Picture Experts Group (MPEG), part of a joint technical committee of the International Standards Organization (ISO) and the International Electrotechnical Commission (IEC). The MPEG committee has been developing standards for the multiplexed, compressed representation of video and associated audio signals.

35 Video decoders are typically embodied as general or special purpose processors and memory. For a conventional MPEG-2 decoder, two decoded reference frames are typically stored in memory at the same time. Thus, the cost of memory often dominates the cost of the decoding subsystem.

40

Accordingly, the invention provides a method for decoding a digital video sequence comprising the steps of: decoding a first picture in the sequence; compressing the first picture; storing a compressed representation of the picture to memory; decompressing a region of the compressed representation of the first picture; and, responsive to the decompressing, decoding a region of a second picture in the sequence.

In one embodiment, the compressing comprises the step of scaling the picture in at least one of the horizontal and vertical directions to a smaller picture, for example scaling by a factor of two in the horizontal direction. It is also possible to store an enhancement version of the first picture to memory, which can be used for subsequent processing.

In another embodiment, the compressing comprises the steps of: segmenting the picture into regions; performing a linear transformation on each region to produce transform coefficients; and quantising the transform coefficients. Preferably, the linear transformation is a Hadamard transform.

The invention also provides apparatus for decoding a compressed digital video sequence comprising: a motion compensation unit for computing reference regions from reference frames; a reference frame compression engine for compressing reference frames and storing them to memory; and a reference frame decompression engine, for decompressing regions of the reference frames compressed by the reference frame compression engine and providing decompressed regions to the motion compensation unit.

In one preferred embodiment, the reference frame compression engine comprises: a linear transformation unit, comprising a Hadamard transformation unit, for performing linear transformations on regions of reference frames and forming reference frame transform coefficients; and means for quantising the reference frame transform coefficients.

This approach reduces the memory requirements of a decoding subsystem by storing reference frames in compressed form. Thus a reference picture in a sequence is decoded. It is then compressed and stored in memory. When the reference frame is needed for motion compensation, it is decompressed.

Preferred embodiments of the invention will now be described in detail by way of example only with reference to the following drawings:

Figure 1 shows an exemplary pair of Groups of Pictures (GOP's);

Figure 2 shows an exemplary macroblock (MB) subdivision of a picture (for 4:2:0 format);

Figure 3 shows a block diagram of a decoder in accordance with the principles of the present invention;

Figure 4 shows memory usage in a conventional decoder;

Figure 5 shows memory usage according to a first embodiment of the reference frame compression engine of Figure 3;

Figure 6 shows memory usage according to a second embodiment of the reference frame compression engine of Figure 3;

Figure 7 shows a block diagram of an embodiment of the reference frame compression engine of Figure 3;

Figure 8 shows a block diagram of an embodiment of the reference frame decompression engine of Figure 3;

Figure 9 shows memory usage of the reference frame compression engine of Figure 7;

Figure 10 is a flow chart of a decoding method in accordance with the principles of the present invention;

Figure 11 is a block diagram of a conventional decoder; and

Figure 12 is a more detailed flow chart showing an embodiment of the decoding method of Figure 10.

THE MPEG-2 ENVIRONMENT

As the present invention may be applied in connection with an MPEG-2 decoder, some pertinent aspects of the MPEG-2 compression algorithm will be reviewed. It is to be noted, however, that the

invention can also be applied to other video coding algorithms which share some of the features of the MPEG-2 algorithm.

To begin with, it will be understood that the compression of a data object, such as a page of text, an image, a segment of speech, or a video sequence, can be thought of as a series of steps, including: 1) a decomposition of that object into a collection of tokens; 2) the representation of those tokens by binary strings which have minimal length in some sense; and 3) the concatenation of the strings in a well-defined order. Steps 2 and 3 are lossless, i.e., the original data is faithfully recoverable upon reversal, and Step 2 is known as entropy coding. Step 1 can be either lossless or lossy in general. Most video compression algorithms are lossy because of stringent bit-rate requirements. A successful lossy compression algorithm eliminates redundant and irrelevant information, allowing relatively large errors where they are not likely to be visually significant and carefully representing aspects of a sequence to which the human observer is very sensitive. The techniques employed in the MPEG-2 algorithm for Step 1 can be described as predictive/interpolative motion-compensated hybrid DCT/DPCM coding. Huffman coding, also known as variable length coding, is used in Step 2.

The MPEG-2 video standard specifies a coded representation of video for transmission as set forth in ISO-IEC JTC1/SC29/WG11, Generic Coding of Moving Pictures and Associated Audio Information: Video, International Standard, 1994. The algorithm is designed to operate on interlaced or noninterlaced component video. Each picture has three components: luminance (Y), red color difference (Cr), and blue color difference (Cb). The video data may be coded in 4:4:4 format, in which case there is one Cr and one Cb sample for each Y sample, in 4:2:2 format, in which case there are half as many Cr and Cb samples as luminance samples in the horizontal direction, or in 4:2:0 format, in which case there are half as many Cr and Cb samples as luminance samples in both the horizontal and vertical directions.

An MPEG-2 data stream consists of a video stream and an audio stream which are packed, together with systems information and possibly other bitstreams, into a systems data stream that can be regarded as layered. Within the video layer of the MPEG-2 data stream, the compressed data is further layered. A description of the organization of the layers will aid in understanding the invention. These layers of the MPEG-2 video

Layered Structure are shown in Figures 1-2. The layers pertain to the operation of the compression algorithm as well as the composition of a compressed bit stream. The highest layer is the Video Sequence Layer, containing control information and parameters for the entire sequence. At the next layer, a sequence is subdivided into sets of consecutive pictures, each known as a "Group of Pictures" (GOP). A general illustration of this layer is shown in Figure 1. Decoding may begin at the start of any GOP, essentially independent of the preceding GOP's. There is no limit to the number of pictures which may be in a GOP, nor do there have to be equal numbers of pictures in all GOP's.

The third or Picture layer is a single picture. A general illustration of this layer is shown in Figure 2. The luminance component of each picture is subdivided into 16 X 16 regions; the color difference components are subdivided into appropriately sized blocks spatially co-sited with the 16 X 16 luminance regions; for 4:4:4 video, the color difference components are 16 X 16, for 4:2:2 video, the color difference components are 8 X 16, and for 4:2:0 video, the color difference components are 8 X 8. Taken together, these co-sited luminance region and color difference regions make up the fifth layer, known as a "macroblock" (MB). Macroblocks in a picture are numbered consecutively in lexicographic order, starting with Macroblock 1.

Between the Picture and MB layers is the fourth or "slice" layer. Each slice consists of some number of consecutive MB's. Finally, each MB consists of four 8 X 8 luminance blocks and 8, 4, or 2 (for 4:4:4, 4:2:2 and 4:2:0 video) chrominance blocks. The Sequence, GOP, Picture, and slice layers all have headers associated with them. The headers begin with byte-aligned Start Codes and contain information pertinent to the data contained in the corresponding layer.

A picture can be either field-structured or frame-structured. A frame-structured picture contains information to reconstruct an entire frame, i.e., the combination of one field containing the odd lines and the other field containing the even lines. A field-structured picture contains information to reconstruct one field. If the width of each luminance frame (in picture elements or pixels) is denoted as C and the height as R (C is for columns, R is for rows), a frame-structured picture contains information for C X R pixels and a field-structured picture contains information for C X R/2 pixels.

The two fields in a frame are the top field and the bottom field. If we number the lines in a frame starting from 1, then the top field contains the odd lines (1, 3, 5, ...) and the bottom field contains the even lines (2, 4, 6, ...). Thus we may also call the top field the odd field and we may also call the bottom field the even field.

A macroblock in a field-structured picture contains a 16 X 16 pixel segment from a single field. A macroblock in a frame-structured picture contains a 16 X 16 pixel segment from the frame that both fields compose; each macroblock contains a 16 X 8 region from each of the two fields.

Within a GOP, three types of pictures can appear. The distinguishing difference among the picture types is the compression method used. The first type, Intraframe pictures or I-pictures, are compressed independently of any other picture. Although there is no fixed upper bound on the distance between I-pictures, it is expected that they will be interspersed frequently throughout a sequence to facilitate random access and other special modes of operation. Predictively motion-compensated pictures (P pictures) are reconstructed from the compressed data in that picture plus two reconstructed fields from previously displayed I or P pictures. Bidirectionally motion-compensated pictures (B pictures) are reconstructed from the compressed data in that picture plus two reconstructed fields from previously displayed I or P pictures and two reconstructed fields from I or P pictures that will be displayed in the future. Because reconstructed I or P pictures can be used to reconstruct other pictures, they are called reference pictures.

With the MPEG-2 standard, a frame can be coded either as a frame-structured picture or as two field-structured pictures. If a frame is coded as two field-structured pictures, then both fields can be coded as a I pictures, the first field can be coded as an I picture and the second field as P picture, both fields can be coded as P pictures, or both fields can be coded as B pictures.

If a frame is coded as a frame-structured I picture, as two field-structured I pictures, or as a field-structured I picture followed by a field-structured P picture, we say that the frame is an I frame; it can be reconstructed without using picture data from previous frames. If a frame is coded as a frame-structured P picture or as two field-structured P pictures, we say that the frame is a P frame; it can be reconstructed from information in the current frame and the previously coded I or P

frame. If a frame is coded as a frame-structured B picture or as two field-structured B pictures, we say that the frame is a B frame; it can be reconstructed from information in the current frame and the two previously coded I or P frames (i.e., the I or P frames that will appear before and after the B frame). We refer to I or P frames as reference frames.

A common compression technique is transform coding. In MPEG-2 and several other compression standards, the discrete cosine transform (DCT) is the transform of choice. The compression of an I-picture is achieved by the steps of 1) taking the DCT of blocks of pixels, 2) quantising the DCT coefficients, and 3) Huffman coding the result. In MPEG-2, the DCT operation converts a block of $n \times n$ pixels into an $n \times n$ set of transform coefficients. Like several of the international compression standards, the MPEG-2 algorithm uses a DCT block size of 8×8 . The DCT transformation by itself is a lossless operation, which can be inverted to within the precision of the computing device and the algorithm with which it is performed.

The second step, quantisation of the DCT coefficients, is the primary source of lossiness in the MPEG-2 algorithm. Denoting the elements of the two-dimensional array of DCT coefficients by C_{mn} , where m and n can range from 0 to 7, aside from truncation or rounding corrections, quantisation is achieved by dividing each DCT coefficient C_{mn} by W_{mn} times QP , with W_{mn} being a weighting factor and QP being the quantiser parameter. The weighting factor W_{mn} allows coarser quantisation to be applied to the less visually significant coefficients. The quantiser parameter QP is the primary means of trading off quality vs. bit-rate in MPEG-2. It is important to note that QP can vary from MB to MB within a picture.

Following quantisation, the DCT coefficient information for each MB is organized and coded, using a set of Huffman codes. As the details of this step are not essential to an understanding of the invention and are generally understood in the art, no further description will be offered here.

Most video sequences exhibit a high degree of correlation between consecutive pictures. A useful method to remove this redundancy prior to coding a picture is "motion compensation". MPEG-2 provides tools for several methods of motion compensation.

The methods of motion compensation have the following in common. For each macroblock, one or more motion vectors are encoded in the bit stream. These motion vectors allow the decoder to reconstruct a macroblock, called the predictive macroblock. The encoder subtracts the "predictive" macroblock from the macroblock to be encoded to form the "difference" macroblock. The encoder uses tools to compress the difference macroblock that are essentially similar to the tools used to compress an intra macroblock.

The type of a picture determines the methods of motion compensation that can be used. The encoder chooses from among these methods for each macroblock in the picture. If no motion compensation is used, the macroblock is intra (I). The encoder can make any macroblock intra. In a P or a B picture, forward (F) motion compensation can be used; in this case, the predictive macroblock is formed from data in the previous I or P frame. In a B picture, backward (B) motion compensation can also be used; in this case, the predictive macroblock is formed from data in the future I or P frame. In a B picture, forward/backward (FB) motion compensation can also be used; in this case, the predictive macroblock is formed from data in the previous I or P frame and the future I or P frame.

Because I and P pictures are used as references to reconstruct other pictures (B and P pictures) they are called reference pictures. Because two reference frames are needed to reconstruct B frames, MPEG-2 decoders typically store two decoded reference frames in memory. The reference frame memory usage for conventional decoders is shown in Figure 4, where we have drawn the frames with height H and width W.

Aside from the need to code side information relating to the MB mode used to code each MB and any motion vectors associated with that mode, the coding of motion-compensated macroblocks is very similar to that of intramode MBs. Although there is a small difference in the quantisation, the model of division by Wmn times QP still holds.

The MPEG-2 algorithm can be used with fixed bit-rate transmission media. However, the number of bits in each picture will not be exactly constant, due to the different types of picture processing, as well as the inherent variation with time of the spatio-temporal complexity of the scene being coded. The MPEG-2 algorithm uses a buffer-based rate control strategy to put meaningful bounds on the variation allowed in the bit-

rate. A Video Buffer Verifier (VBV) is devised in the form of a virtual buffer, whose sole task is to place bounds on the number of bits used to code each picture so that the overall bit-rate equals the target allocation and the short-term deviation from the target is bounded. This rate control scheme can be explained as follows. Consider a system consisting of a buffer followed by a hypothetical decoder. The buffer is filled at a constant bit-rate with compressed data in a bit stream from the storage medium. Both the buffer size and the bit-rate are parameters which are transmitted in the compressed bit stream. After an initial delay, which is also derived from information in the bit stream, the hypothetical decoder instantaneously removes from the buffer all of the data associated with the first picture. Thereafter, at intervals equal to the picture rate of the sequence, the decoder removes all data associated with the earliest picture in the buffer.

Figure 11 shows a diagram of a conventional video decoder. The compressed data enters as signal 1101 and is stored in the compressed data memory 1102. The variable length decoder 1104 reads the compressed data as signal 1103 and sends motion compensation information as signal 1108 to the motion compensation unit 1109 and quantised coefficients as signal 1107 to the inverse quantisation unit 1110. The motion compensation unit reads the reference data from the reference frame memory 1112 as signal 1111 to form the predicted macroblock, which is sent as the signal 1114 to the adder 1117. The inverse quantisation unit computes the unquantised coefficients, which are sent as signal 1113 to the inverse transform unit 1115. The inverse transform unit computes the reconstructed difference macroblock as the inverse transform of the unquantised coefficients. The reconstructed difference macroblock is sent as signal 1116 to the adder 1117, where it is added to the predicted macroblock. The adder 1117 computes the reconstructed macroblock as the sum of the reconstructed difference macroblock and the predicted macroblock. The reconstructed macroblock is then sent as signal 1118 to the demultiplexor 1119, which stores the reconstructed macroblock as signal 1121 to the reference memory if the macroblock comes from a reference picture or sends it out (to memory or display) as signal 1120. Reference frames are sent out as signal 1124 from the reference frame memory.

PREFERRED EMBODIMENT OF A DECODER

A decoding method in accordance with the principles of the present invention will now be described. Reference pictures are stored in memory in compressed form. The compression method used can be lossy or lossless and is preferably simpler than, and therefore different from, the compression used to originally compress the video. In embodiments where the compression is lossy, the decoding method will be inexact, thus, the decoded output video signal will typically differ from the output signal of a conventional video decoder.

The steps involved in decoding are shown in Figure 10. In step 1001, a picture is decoded, and data needed for motion compensation is decompressed before being used. Step 1002 checks if the decoded picture was a reference (I or P) picture; if it was, control moves to step 1003; otherwise, control moves to step 1004. In step 1003, the reference picture is compressed and stored to memory, and control then moves to step 1004. In step 1004, we go to the next picture and control returns to step 1001.

A block diagram of a decoder according to an embodiment of the invention is shown in Figure 3. The decoder of Figure 3 is preferably embodied as an application specific integrated circuit (ASIC) connected to one or more memory devices. The compressed data enters as signal 301 and is stored in the compressed data memory 302. The variable length decoder 304 reads the compressed data as signal 303 and sends motion compensation information as signal 308 to the motion compensation unit 309 and quantised coefficients as signal 307 to the inverse quantisation unit 310. The reference frame decompression engine 327 reads compressed reference frame data signal 326 from the reference frame memory 312, decompresses the data, and sends the decompressed reference frame data as signal 311 to the motion compensation unit. The motion compensation unit uses signals 311 and 308 to form the predicted macroblock, which is sent as the signal 314 to the adder 317. The inverse quantisation unit computes the unquantised coefficients, which are sent as signal 313 to the inverse transform unit 315. The inverse transform unit computes the reconstructed difference macroblock as the inverse transform of the unquantised coefficients. The reconstructed difference macroblock is sent as signal 316 to the adder 317, where it is added to the predicted macroblock. The adder 317 computes the reconstructed macroblock as the sum of the reconstructed difference macroblock and the predicted macroblock. The reconstructed macroblock is then sent as signal 318 to the demultiplexor 319, which sends the reconstructed macroblock as signal

321 to the reference frame compression engine 327 if the macroblock comes from a reference picture or sends the data out as signal 320 if the macroblock comes from a B picture. The reference frame compression engine 328 compresses the reconstructed macroblock (signal 321) and stores the compressed version of the macroblock as signal 325 in the reference frame memory. Reference data is read out as signal 324 after being decompressed by the reference frame decompression engine.

In a first embodiment of a reference frame compression engine 328, each reference frame is scaled to a smaller version of the frame and stored in memory. For example, each frame could be scaled by a factor of two horizontally and not at all vertically. This scaling is illustrated in Figure 5. Note that for this example only half of the memory used by a conventional decoder for reference frame storage is needed.

For the first embodiment of a reference frame compression engine, the reference frame decompression engine 327 scales the reference frame back to full size. For example, if the reference frame compression engine 328 scales by a factor of two horizontally and not at all vertically, the reference frame decompression engine 327 could repeat pixels in the scaled frame to scale back to full size.

In a second embodiment of a reference frame compression engine, each reference frame is scaled to a smaller version of the frame and stored in memory, and an enhancement version of the reference frame is also stored in memory. This enhancement version is used together with the scaled version for motion compensation of P pictures. When a P picture is decoded the enhancement version of the previous reference frame is overwritten when it is no longer needed for motion compensation for the P picture being decoded. This means that when a B picture is decoded the scaled version and enhancement version of the future frame will be available for motion compensation but only the scaled version of the previous frame will be available. For example, each frame could be scaled by a factor of two horizontally and not at all vertically to create the scaled version. This scaling is done by discarding every other pixel horizontally. The discarded pixels are used as the enhancement version. In this case, by using both the enhancement version and the scaled version the frame can be reconstructed exactly. This means that P pictures (and I pictures) will be reconstructed exactly but B pictures will not. The memory allocation for this embodiment is illustrated in Figure 6.

For this embodiment of a reference frame compression engine, the reference frame decompression engine works by scaling the reference frame back to full size, using only the scaled version if only that version is stored in memory, but both the scaled and enhancement versions if both are available.

The operation of a decoder using the second embodiment of a reference frame scaling engine is described by the flow chart shown in Figure 12. Step 1201 checks if a picture to be decoded is a B picture; if it is, control goes to step 1202, otherwise control goes to step 1203. Step 1202 decodes the B picture using the scaled previous frame and the scaled and enhanced future frame for motion compensation. Step 1203 decodes a reference picture using the scaled and enhanced versions of the previous frame. After step 1203, control goes to step 1204, which stores the scaled and enhanced version of the picture to memory; the enhanced version overwrites data in the previous frame. After step 1204 or step 1202, control goes to step 1205, which moves the decoding to the next picture. Control then returns to step 1201.

A third embodiment of a reference frame compression engine is shown in Figure 7. The reference data is segmented into blocks, and these blocks are then sent as signal 701 to a Hadamard transform unit 702. A definition of the Hadamard transform can be found in "Digital Image Processing" by C. Gonzalez and P. Wintz, second edition, 1987; section 3.5.2. In this embodiment, data is segmented into 4 X 1 blocks and it is then subjected to a 4 X 1 Hadamard transform. Denoting the inputs to a 4 X 1 Hadamard transform as x_0 , x_1 , x_2 , and x_3 and the outputs as y_0 , y_1 , y_2 , and y_3 , the outputs can be computed from the inputs as:

$$\begin{aligned} y_0 &= x_0 + x_1 + x_2 + x_3 \\ y_1 &= x_0 + x_1 - x_2 - x_3 \\ y_2 &= x_0 - x_1 - x_2 + x_3 \\ y_3 &= x_0 - x_1 + x_2 - x_3 \end{aligned}$$

The Hadamard coefficients are sent as signal 703 to the divide and round unit 704, which divides each coefficient and rounds to the nearest integer. In this embodiment, coefficient y_0 is divided by 4 and the other coefficients are divided by 8. The rounded coefficients are sent as signal 705 to the clipping unit 706, which clips each coefficient to an interval, and are outputted as signal 707. For this embodiment, the coefficient y_0 is clipped to the interval [0, 255], the coefficient y_1 is

clipped to the interval $[-32, 31]$, and the coefficients y_2 and y_3 are clipped to the interval $[-16, 15]$. Clipping a coefficient to the interval $[A, B]$ means that it is replaced with A if it is less than A , replaced with B if it is greater than B , and unchanged otherwise. Note that because y_0 is an integer in $[0, 255]$, it can be represented with 8 bits, because y_1 is an integer in $[-32, 31]$, it can be represented with 6 bits, and because y_2 and y_3 are integers in $[-16, 15]$ they can be represented with 5 bits each. Thus y_0 , y_1 , y_2 and y_3 can be represented with a total of $8 + 6 + 5 + 5 = 24$ bits. For this embodiment, the input data (x_0 , x_1 , x_2 , and x_3) are 8 bit numbers, so the compression ratio is $4 \times 8 : 24 = 4 : 3$. The memory usage for this embodiment is shown in Figure 9, where it is shown that each compressed row uses $3/4$ the storage of an uncompressed row.

An embodiment of a reference frame decompression engine suitable for use in the decoder of Figure 3 when the reference frame compression engine of Figure 7 is used is shown in Figure 8. The compressed reference frame data is sent as signal 804 to the multiplier 801. In this embodiment, the first coefficient in each 4×1 block is multiplied by 1 and the others by two. These are then sent as signal 805 to the Hadamard transform unit 802, which computes the Hadamard transform on each 4×1 block. The transformed data is then sent as signal 806 to the clipping unit 803, which clips each input to $[0, 255]$, and sends out the clipped data as signal 807.

CLAIMS

1. A method for decoding a digital video sequence comprising the steps of:

5 decoding a first picture in the sequence;

compressing the first picture;

10 storing a compressed representation of the picture to memory;

decompressing a region of the compressed representation of the first picture; and,

15 responsive to the decompressing, decoding a region of a second picture in the sequence.

2. The method of Claim 1 wherein the compressing comprises the step of scaling the picture in at least one of the horizontal and vertical directions to a smaller picture.

3. The method of Claim 2, wherein the scaling comprises scaling by a factor of two in the horizontal direction.

25 4. The method of any preceding Claim, further comprising the step of storing an enhancement version of the first picture to memory.

5. The method of Claim 1 wherein the compressing comprises the steps of:

30 segmenting the picture into regions;

performing a linear transformation on each region to produce transform coefficients; and

35 quantising the transform coefficients.

6. The method of Claim 5 wherein the linear transformation is a Hadamard transform.

7. Apparatus for decoding a compressed digital video sequence comprising:

5 a motion compensation unit (309) for computing reference regions from reference frames;

a reference frame compression engine (328) for compressing reference frames and storing them to memory (312); and

10 a reference frame decompression engine (327), for decompressing regions of the reference frames compressed by the reference frame compression engine and providing decompressed regions to the motion compensation unit.

15 8. The apparatus of Claim 7 wherein the reference frame compression engine comprises means for scaling reference frames and storing them to memory.

20 9. The apparatus of Claim 7 or 8 wherein the reference frame compression engine comprises means for storing an enhancement version of the compressed reference frame to memory.

25 10. The apparatus of Claim 7 wherein the reference frame compression engine comprises:

a linear transformation unit (802), for performing linear transformations on regions of reference frames and forming reference frame transform coefficients; and

30 means for quantising the reference frame transform coefficients;

35 11. The apparatus of Claim 10 wherein the linear transformation unit is a Hadamard transformation unit.



The Patent Office

16

Application No: GB 9701471.6
Claims searched: 1 to 11

Examiner: John Donaldson
Date of search: 17 March 1997

Patents Act 1977 Search Report under Section 17

Databases searched:

UK Patent Office collections, including GB, EP, WO & US patent specifications, in:

UK Cl (Ed.O): H4F(FRC, FRD, FRG, FRM, FRP, FRR, FRS, FRT, FRW, FRX)

Int Cl (Ed.6): H04N 7/00, 7/24, 7/26, 7/30, 7/32, 7/34, 7/36, 7/48, 7/50, 11/00,
11/02, 11/04

Other: Online:WPI

Documents considered to be relevant:

Category	Identity of document and relevant passage	Relevant to claims
A	GB 2295255 A (IBM), see abstract	-

X Document indicating lack of novelty or inventive step
Y Document indicating lack of inventive step if combined with one or more other documents of same category.

& Member of the same patent family

A Document indicating technological background and/or state of the art.
P Document published on or after the declared priority date but before the filing date of this invention.
E Patent document published on or after, but with priority date earlier than, the filing date of this application.

**This Page is Inserted by IFW Indexing and Scanning
Operations and is not part of the Official Record**

BEST AVAILABLE IMAGES

Defective images within this document are accurate representations of the original documents submitted by the applicant.

Defects in the images include but are not limited to the items checked:

☐ **BLACK BORDERS**

☐ **IMAGE CUT OFF AT TOP, BOTTOM OR SIDES**

☐ **FADED TEXT OR DRAWING**

☒ **BLURRED OR ILLEGIBLE TEXT OR DRAWING**

☐ **SKEWED/SLANTED IMAGES**

☐ **COLOR OR BLACK AND WHITE PHOTOGRAPHS**

☐ **GRAY SCALE DOCUMENTS**

☐ **LINES OR MARKS ON ORIGINAL DOCUMENT**

☐ **REFERENCE(S) OR EXHIBIT(S) SUBMITTED ARE POOR QUALITY**

☐ **OTHER:** _____

IMAGES ARE BEST AVAILABLE COPY.

As rescanning these documents will not correct the image problems checked, please do not report these problems to the IFW Image Problem Mailbox.